Available online at www.sciencedirect.com
ScienceDirect
Computers and Mathematics with Applications 55 (2008) 2247–2265
An International Journal
computers & mathematics with applications
www.elsevier.com/locate/camwa

A numerical comparative study on data assimilation using Kalman filters

G. Dimitriu

Department of Mathematics and Informatics, University of Medicine and Pharmacy, Faculty of Pharmacy, 700115 Iaşi, Romania

Abstract

Kalman filtering has become a powerful framework for solving data assimilation problems. Of interest here are the low-rank filters which are computationally efficient for solving large-scale data assimilation problems. Together with theoretical aspects on the basis of which some common low-rank filters are designed, the paper also presents numerically comparative results of data assimilation using an air pollution model. The performance of such filters, as depending on the distance between the measurement locations and emission points, is investigated.

Keywords: Data assimilation; Kalman filters; Low-rank filters; Air pollution modeling; Advection–diffusion equation

1. Introduction

Originally designed for guidance problems, the Kalman filter [1] has a long history of merging models and measurements in electrical engineering and control. The growing availability of cheap computing power during the last decade made the filter approach feasible for large air pollution models too. Kalman filtering represents a powerful framework for solving data assimilation problems [2]. For the implementation of a Kalman filter, the evolution of the state and observation of measurements can be described with the stochastic system:

$$\mathbf{x}^t[k+1] = \mathbf{A}[k]\mathbf{x}^t[k] + \eta[k], \qquad \mathbf{y}^o[k] = \mathbf{H}[k]'\mathbf{x}^t[k] + \nu[k] \tag{1}$$

with $\mathbf{x}^t[k] \in \mathbb{R}^n$ being the true state vector at time $t[k]$, $\mathbf{A}[k]$ a deterministic model, $\eta[k] \in \mathbb{R}^n$ a Gaussian distributed model error (zero mean, covariance $\mathbf{Q}$), and $\mathbf{y}^o[k] \in \mathbb{R}^r$ a vector of observations with $\nu[k]$ the representation error (Gaussian with zero mean and covariance $\mathbf{R}$). The superscripts t, o, and later on f and a refer to the true, observed, forecasted and analyzed entities, respectively. We also mention that the time indices for $\mathbf{A}$ and $\mathbf{H}'$ will be omitted in the following equations, assuming that the time is implied by the state where the operators act on.

The aim of the filter operations is to obtain the mean $\hat{\mathbf{x}}^a$ and covariance $\mathbf{P}^a$ for the probability density of the true state. The filter equations for this system contain the *forecast* stage given by:

$$\hat{\mathbf{x}}^f[k+1] = \mathbf{A}[k]\hat{\mathbf{x}}^a[k], \qquad \mathbf{P}^f[k+1] = \mathbf{A}\mathbf{P}^a[k]\mathbf{A} + \mathbf{Q}[k] \tag{2}$$

E-mail address: dimitriu@umfiasi.ro.

0898-1221/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2007.11.009

and *analysis* stage expressed by

$$\mathbf{x}^a = \hat{\mathbf{x}}^f + \mathbf{K}(\mathbf{y}^o - \mathbf{H}'\hat{\mathbf{x}}^f) \tag{3}$$

$$\mathbf{P}^a = \begin{cases} (\mathbf{I} - \mathbf{K}^{MV}\mathbf{H}')\mathbf{P}^f, & \mathbf{K}^{MV} = \mathbf{P}^f\mathbf{H}'(\mathbf{H}'\mathbf{P}^f\mathbf{H} + \mathbf{R})^{-1} \\ (\mathbf{I} - \mathbf{K}\mathbf{H}')\mathbf{P}^f(\mathbf{I} - \mathbf{K}\mathbf{H}')' + \mathbf{K}\mathbf{R}\mathbf{K}', & \text{arbitrary gain } \mathbf{K}. \end{cases} \tag{4}$$

In the case of a large model, the propagation of the covariance matrix in (2) represents the most expensive part in the full rank filter. If $\mathbf{A}$ is defined by an $n \times n$ matrix, then the dynamical model is called $2n$ times to perform the operation $\mathbf{A}(\mathbf{A}\mathbf{P})'$. Limitation of both the number of model evaluations as well as the storage requirements will be achieved in this study by reducing the rank of the covariance matrix.

Bierman [3] proposed to write the equations for the Kalman filter using the factorization $\mathbf{P} = \mathbf{S}\mathbf{S}'$. Numerical inaccuracies made in the computation and storage of the matrix $\mathbf{S}$ will never affect the property of the positive definiteness of $\mathbf{P}$. Inaccuracies will even be reduced, since the condition number of $\mathbf{S}$ is only the square root of the condition number of $\mathbf{P}$.

The idea of factorization is useful to reduce the storage requirements of $\mathbf{P}$. Consider a covariance matrix $\mathbf{P}$ written as the product of a rectangular matrix square root $\mathbf{S}$ and its transpose:

$$\mathbf{P}_{n\times n} = \mathbf{S}_{n\times m}\mathbf{S}'_{m\times n}.$$

In order to obtain the Kalman filter in square root form, apart from the previous factorization $\mathbf{P} = \mathbf{S}\mathbf{S}'$ for the covariance of the true state, we also introduce the factorizations $\mathbf{Q} = \mathbf{T}\mathbf{T}'$ and $\mathbf{R} = \mathbf{U}\mathbf{U}'$ for the covariance of the forecast and representation error, respectively. Further, a matrix $\Psi' = \mathbf{H}'\mathbf{S}$ is introduced for the mapping of the forecast covariance root to the observation space.

After (2), the forecasts of mean and covariance become:

$$\hat{\mathbf{x}}^f[k+1] = \mathbf{A}\hat{\mathbf{x}}^a[k] \tag{5}$$

$$(\mathbf{S}^f\mathbf{S}^{f\prime})[k+1] = \mathbf{A}(\mathbf{S}^a\mathbf{S}^{a\prime})[k]\mathbf{A} + \mathbf{T}\mathbf{T}'[k]$$

$$\text{or} \quad \mathbf{S}^f[k+1] = \left[\mathbf{A}\mathbf{S}^a[k], \mathbf{T}[k]\right]. \tag{6}$$

The second formula in (6) is able to reduce both the computational complexity and the numerical inaccuracies, since the condition number of $\mathbf{S}^f$ or $\mathbf{S}^a$ is only the square root of the condition number of $\mathbf{P}^f$ and $\mathbf{P}^a$, respectively. The introduction of a forecast error leads to the extension of the square root with the columns of $\mathbf{T}$. Each new column introduces a new direction for the uncertainty of the state vector. To prevent the number of modes from growing to infinity, filter algorithms based on factorizations include approximations or mechanisms to avoid the growth, for example avoiding the use of dynamic noise completely, projection of $\mathbf{T}$ on the base spanned by $\mathbf{AS}$, or reduction of the number of columns whenever necessary. If $\mathbf{T}$ is to be added to the covariance square root, the degree of freedom in the system noise (rank of $\mathbf{T}$) should be of order 10–100 to keep the storage and propagation of the covariance square root within feasible bounds.

This paper presents mathematical aspects of some Kalman filters in factorized form, together with comparative numerical results obtained by applying such filters to data assimilation problems. The paper is organized as follows. In Section 2 we briefly describe some factorized filters, namely the: Reduced Rank Square Root (RRSQRT) filter, Partially Orthogonal Ensemble Kalman (POENK) filter and its variant (COFFEE), also including the Ensemble Kalman filter. In Section 3, the performance of the various algorithms is illustrated by numerical tests carried out with an advection–diffusion model application. The last section contains some concluding remarks.

2. Description of some factorized filters

2.1. RRSQRT filter

In the *Reduced Rank SQuare RooT* (RRSQRT) formulation of the Kalman filter, the covariance matrix is expressed in a limited number of (orthogonal) modes, which are re-orthogonalized and truncated to a fixed number during each

time step. The basic formulation is a direct translation of the linear Kalman filter into its square root formulation, leading to:

$$\hat{\mathbf{x}}^f[k+1] = \mathbf{A}\hat{\mathbf{x}}^a[k], \qquad S^f[k+1] = [AS^a[k], T[k]] \tag{7}$$

$$\Psi = \mathbf{H}'\mathbf{S}^f[k+1], \tag{8}$$

$$\mathbf{K} = \mathbf{S}^f[k+1]\,\Psi[\,\Psi'\Psi + \mathbf{R}[k+1]]^{-1}, \tag{9}$$

$$\hat{\mathbf{x}}^a[k+1] = \hat{\mathbf{x}}^f + \mathbf{K}(\mathbf{y}^o[k+1] - \mathbf{H}'\hat{\mathbf{x}}^f[k+1]), \tag{10}$$

$$\mathbf{S}^a[k+1] = \mathbf{S}^f[k+1][\mathbf{I} - \Psi(\Psi'\Psi + \mathbf{R}[k+1])^{-1}\Psi']^{\frac{1}{2}}, \tag{11}$$

$$\mathbf{V}\Lambda\mathbf{V}' = \mathbf{S}^a[k+1]'\mathbf{S}^a[k+1], \qquad \tilde{\mathbf{S}}^a[k+1] = \mathbf{S}^a[k+1]\tilde{\mathbf{V}}. \tag{12}$$

Generally, the algorithm is initialized with an empty covariance square root; new columns are added at every time step due to the introduction of system noise $\mathbf{T}$ in (7). For each of the m modes stored in $\mathbf{S}$, the forecast of the covariance requires one evaluation of the model $\mathbf{A}$. The analysis steps (9)–(11) are usually implemented in the form of a sequential update for scalar measurements. An important part of the RRSQRT algorithm is the reduction of the covariance square root (12). With the introduction of system noise in (7), the number of modes has grown from m to $m + q$, where q is the number of columns in $\mathbf{T}$ (rank of $\mathbf{Q}$). The reduction step reduces the size to m again. Matrix $\tilde{\mathbf{V}}$ contains the eigenvectors of $(\mathbf{S}^a)'\mathbf{S}^a$ corresponding with the largest m eigenvalues. The new matrix $\mathbf{S}^a\tilde{\mathbf{V}}$ represents an approximation of $\mathbf{S}$, maintaining the largest singular vectors. In term of computational costs, the most expensive part of the RRSQRT filter is formed by the propagation of the modes in (7), when for each mode the model should be called once. The reduction should therefore reduce the number of modes as far as possible.

2.2. Ensemble filter

In comparison with the RRSQRT filter, which is based on the factorization of the covariance matrix, the ENsemble Kalman Filter (ENKF) is based on the convergence of large numbers. Both approaches lead to a low-rank approximation of the covariance matrix. The ensemble filter was introduced by Evensen [4] for assimilation of data in oceanographic models.

The essential idea behind the ensemble filter is to express the probability function of the state in an ensemble of possible states $\{\xi_1, \ldots, \xi_N\}$. Each ensemble member is assumed to be a single sample out of the distribution of the true state. Whenever necessary, statistical moments are approximated with sample statistics:

$$\hat{\mathbf{x}} \approx \frac{1}{m}\sum_{j=1}^{m}\xi_j, \qquad \mathbf{P} \approx \frac{1}{m-1}\sum_{j=1}^{m}(\xi_j - \hat{\mathbf{x}})(\xi_j - \hat{\mathbf{x}})', \ldots. \tag{13}$$

The sample statistics will always converge to the true values with increasing ensemble size. However, the convergence is rather slow (order $1/\sqrt{m}$), and this is the only serious disadvantage of the ensemble filter. Evensen [5] stated that for practical ensemble sizes of $\mathcal{O}(100)$, the errors in the filter will be dominated by statistical noise. To remove a part of the statistical noise, Houtekamer and Mitchell [6] used a cutoff radius after which correlations are ignored whenever these are extracted from the ensemble.

An important difference between the pair $(\hat{\mathbf{x}}, P)$ of the Kalman or factorized filter and the ensemble statistics (13) is that the latter are much more connected with each other. In traditional Kalman filters, $\hat{\mathbf{x}}$ and $\mathbf{P}$ are processed more or less independently of each other. The mean $\hat{\mathbf{x}}$ is analyzed using a gain matrix computed from $\mathbf{P}$, but $\mathbf{P}$ is never affected by $\hat{\mathbf{x}}$; the covariance and gain could even be computed off-line.

It is possible to reformulate the ensemble in terms of a (sample) covariance square root:

$$\mathbf{P} = \sum_{k=1}^{m}\mathbf{e}_k\mathbf{e}_k' = \mathbf{E}\mathbf{E}', \qquad \mathbf{e}_k = \frac{\xi_k - \overline{\xi}}{\sqrt{m-1}}. \tag{14}$$

Each ensemble member defines a rank one covariance matrix $\mathbf{e}_k\mathbf{e}_k'$. We notice that at least two ensemble members are required to provide a sample mean and sample covariance. This is no different for the filters based on factorizations,

which require at least two states for the mean and covariance too: the mean itself and one mode for a rank-one covariance matrix.

The filter equations for the ensemble filter are different from the previously described factorized filters in their operating on an ensemble of states instead of a mean and covariance factor. Given an initial ensemble of states describing a range of possible true states, a forecast of the statistics for the true state at a future time is simply obtained from propagated ensemble members. In case of a non-linear model, the propagation becomes:

$$\xi_k^f[k+1] = \mathbf{M}[\xi_k^a[k]) + \eta_k[k], \qquad \eta_k[k] \sim \mathcal{N}(0, \mathbf{Q}[k]), \tag{15}$$

where a sample of the system noise is obtained from a random generator. Whenever measurements are available, each of the ensemble members is analyzed with a linear gain:

$$\xi_j^a[k+1] = \xi_j^f[k+1] + \mathbf{K}(\mathbf{y}^o[k+1] + \nu_j - \mathbf{H}'\xi^f[k+1]), \tag{16}$$

where $\nu_j \sim \mathcal{N}(0, \mathbf{R}[k+1])$. The vectors ν_j denote samples of the representation error, drawn from a random generator. With $\mathbf{P}^e$ and $\mathbf{R}^e$ being the sample covariances of the vectors ξ_j and ν_j respectively, this analysis scheme leads to an analyzed mean and covariance given by (a bar denotes an ensemble mean):

$$\begin{aligned}
\hat{\mathbf{x}}^a &= \overline{\xi_j^a} = \overline{\xi_j^f} + \mathbf{K}(\mathbf{y}^o + \overline{\nu_j} - \mathbf{H}'\overline{\xi_j^f}) \\
\mathbf{P}^{e,a} &= \overline{(\xi_j^a - \overline{\xi_j^a})(\xi_j^a - \overline{\xi_j^a})'} \\
&= [\mathbf{I} - \mathbf{K}\mathbf{H}]\mathbf{P}^{e,f}[\mathbf{I} - \mathbf{K}\mathbf{H}]' + \mathbf{K}^e\mathbf{R}^e\mathbf{K}' \\
&\quad + \mathcal{O}\left(\overline{(\nu_j - \overline{\nu_j})(\nu_j - \overline{\nu_j})} - \mathbf{R}\right) + \mathcal{O}\left(\overline{(\xi_j^a - \overline{\xi_j^a})(\nu_j - \overline{\nu_j})}\right).
\end{aligned}$$

The last two terms converge to zero with order $1/\sqrt{m}$. If these terms are omitted, the analysis scheme produces what is expected from (4) for the analysis of covariance $\mathbf{P}^e$ with an arbitrary gain matrix $\mathbf{K}$. The ensemble analysis (16) is independent of the gain matrix used. Under the assumption that the probability densities of both state and measurements are close to Gaussian, a gain matrix for the ensemble filter might be formed using the ensemble covariance:

$$\mathbf{K}^e = \mathbf{P}^e\mathbf{H}\,[\mathbf{H}'\mathbf{P}^e\mathbf{H} + \mathbf{R}]^{-1}. \tag{17}$$

2.3. *Hybrid approaches:* POENK *and* COFFEE *filters*

A new direction in the implementation of low-rank filters is the use of two filters next to each other. The combination should compensate for errors made in one or both of the individual filters.

The *Partially Orthogonal ENsemble Kalman* filter (POENK) runs a RRSQRT filter next to an ENKF. The basic idea is to let the RRSQRT part compute the bulk of the covariance structure, as described in the first modes. The ENKF part should account for the truncation error, by introducing directions in the covariance matrix that have been lost during the reduction. This procedure incorporates the advantages of both filter types, and accounts for their major disadvantages. Ensemble filters suffer from a lack of convergence; many ensembles are required before sample mean and correlations are stable. An ensemble filter is able to estimate and maintain any correlation introduced by the stochastic model, however. The reverse holds for the RRSQRT filter; a few modes are sufficient to describe the main part of the covariance structure, but some of the correlation structure is lost during the reduction.

The gain matrix used in the POENK filter is computed with a covariance matrix $\mathbf{P}^{\text{poen}}$ formed from the covariances in the two underlying filters. The bulk of $\mathbf{P}^{\text{poen}}$ is obtained from covariances $\mathbf{P}^{rr}$ of the RRSQRT part, and the remainder from a projection of the ensemble covariance on the orthogonal complement of $\mathbf{P}^{rr}$:

$$\mathbf{P}^{\text{poen}} = \mathbf{P}^{rr} + \mathbf{P}^{en\perp rr}, \qquad \mathbf{K}^{\text{poen}} = \mathbf{P}^{\text{poen}}\mathbf{H}(\mathbf{H}'\mathbf{P}^{\text{poen}}\mathbf{H} + \mathbf{R})^{-1}. \tag{18}$$

The gain in (18) is used to analyze both the $\hat{\mathbf{x}}^f$ and $\mathbf{S}^f$ of the RRSQRT part, and the ensemble members in the ENKF part. The new gain matrix $\mathbf{K}^{\text{poen}}$ acts as a variance reducer for the enesemble, since the ensemble mean is less sensitive to fluctuations due to small ensemble sizes [7]. It is efficiently computed using the square root $\mathbf{S}^{\text{poen}}$ of $\mathbf{P}^{\text{poen}}$.

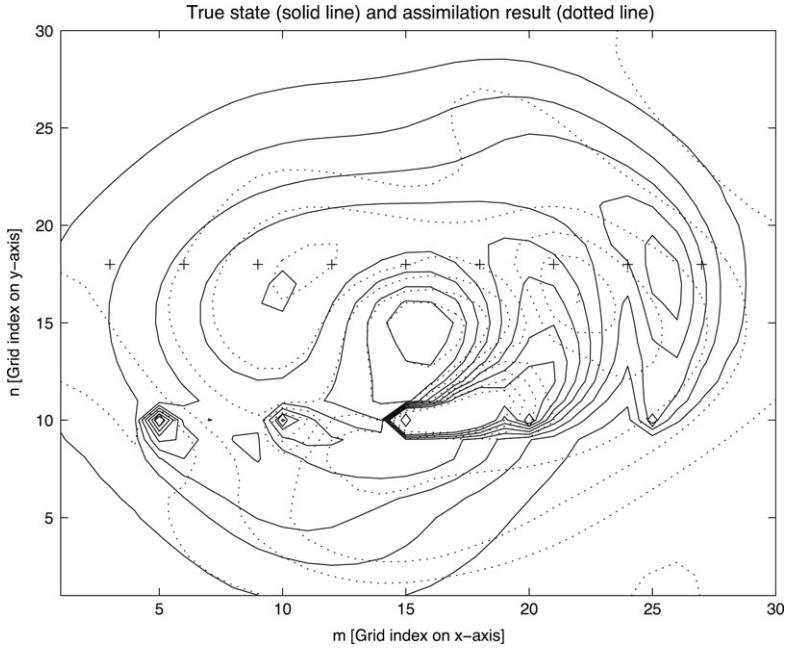

Fig. 1. The concentrations calculated using RRSQRT filter with $(q, N) = (5, 30)$ at time step $k = 100$ (Case I).

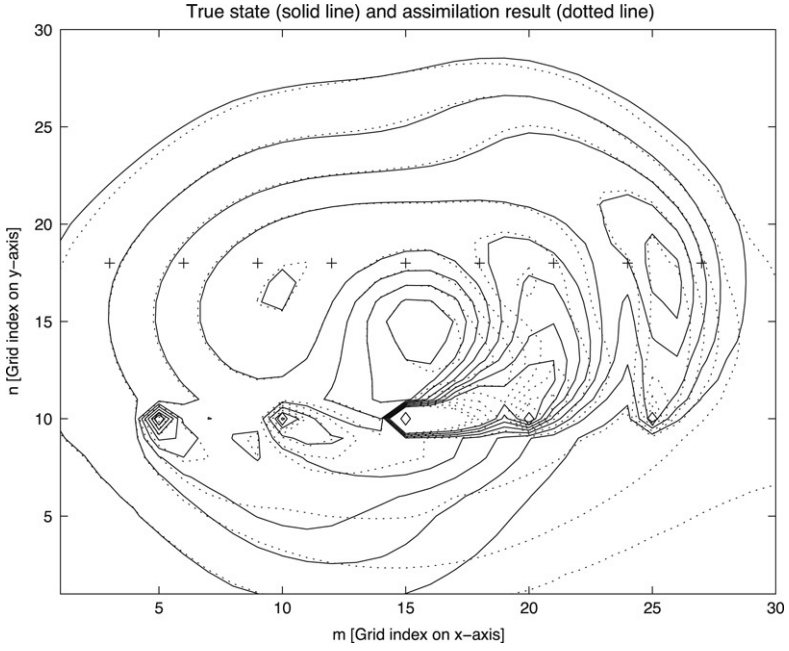

Fig. 2. The concentrations calculated using RRSQRT filter with $(q, N) = (25, 30)$ at time step $k = 100$ (Case I).

Expressed in the square root $\mathbf{S}$ from the RRSQRT part and the 'ensemble square root' $\mathbf{E}$ defined in (14), the square root $\mathbf{S}^{\text{poen}}$ is computed from:

$$\mathbf{\Pi}^{\|} = \mathbf{S}(\mathbf{S}'\mathbf{S})^{-1}\mathbf{S}', \qquad \mathbf{E}^{\|} = \mathbf{\Pi}^{\|}\,\mathbf{E}, \qquad \mathbf{E}^{\perp} = \mathbf{E} - \mathbf{E}^{\|}, \qquad \mathbf{S}^{\text{poen}} = \left[\,\mathbf{S}^{rr},\ \mathbf{E}^{\perp}\,\right],$$

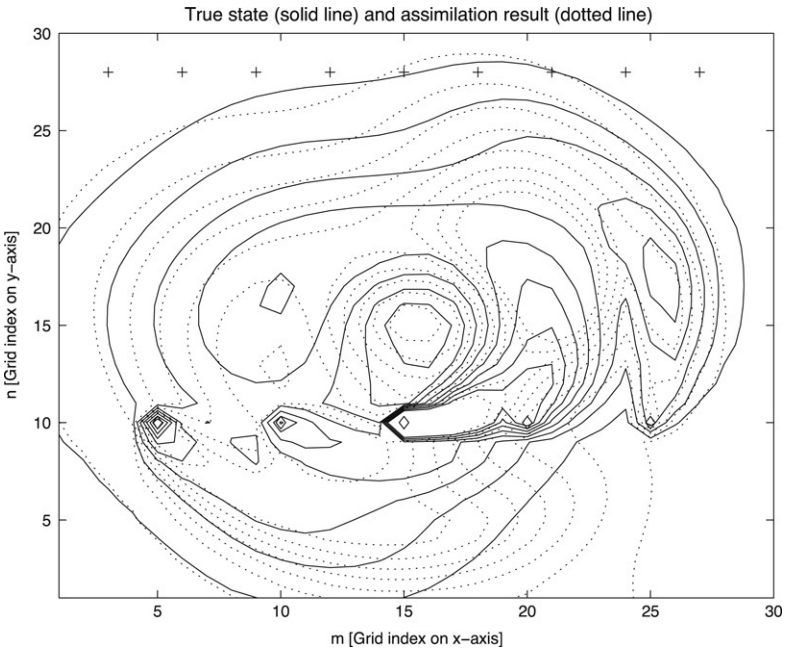

Fig. 3. The concentrations calculated using RRSQRT filter with $(q, N) = (5, 30)$ at time step $k = 100$ (Case II).

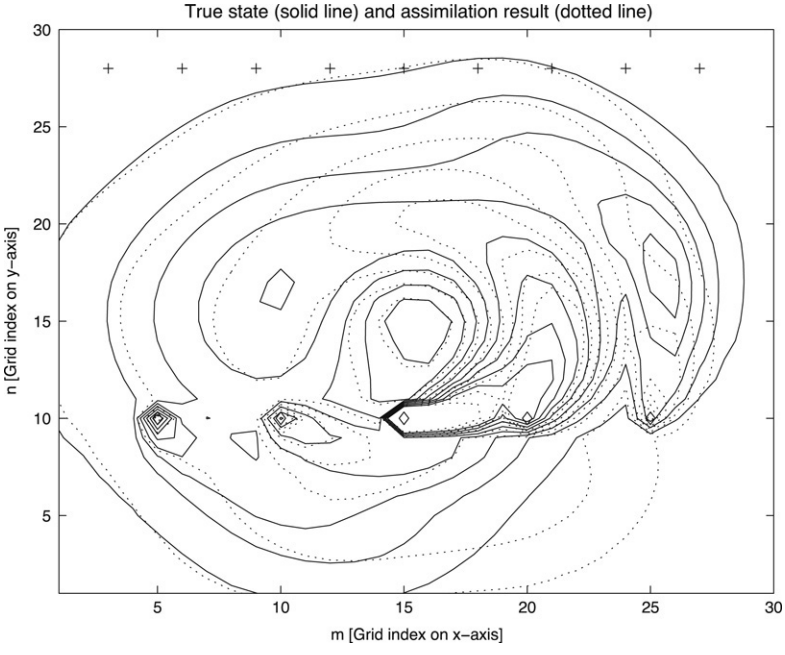

Fig. 4. The concentrations calculated using RRSQRT filter with $(q, N) = (25, 30)$ at time step $k = 100$ (Case II).

where $\mathbf{\Pi}^{\parallel}$ is the projection matrix on the subspace spanned by the columns of $\mathbf{S}$. Thus, the covariance square root of the POENK filter is obtained by adding a number of columns to $\mathbf{S}^{rr}$ equal to the ensemble size.

A variant of POENK filter is the Complementary Orthogonal subspace Filter For Efficient Ensembles (COFFEE) algorithm (see [8] for details).

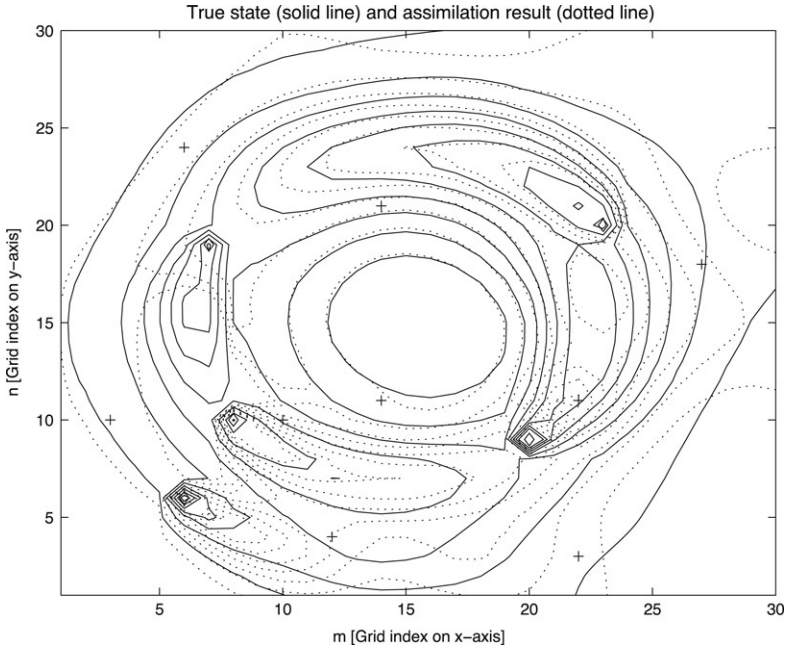

Fig. 5. The concentrations calculated using RRSQRT filter with $(q, N) = (5, 30)$ at time step $k = 100$ (Case III).

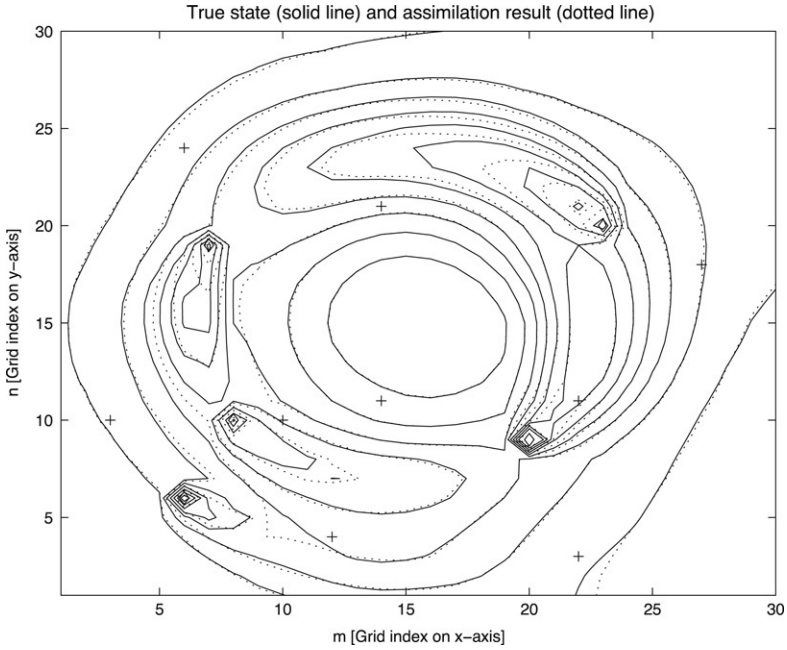

Fig. 6. The concentrations calculated using an RRSQRT filter with $(q, N) = (25, 30)$ at time step $k = 100$ (Case III).

3. Computational issues with an advection–diffusion model

The performances of four types of low-rank filters (RRSQRT, ENK, POENK and COFFEE) were tested during a filter experiment with simulated data. We used some slightly modified versions of Matlab routines originally created by Verlaan et al. [8].

In comparison with the study of Verlaan et al. [8], in which the locations of the measurement and emission points are spread on the whole domain (Case III below), here we also take into consideration other two particular cases.

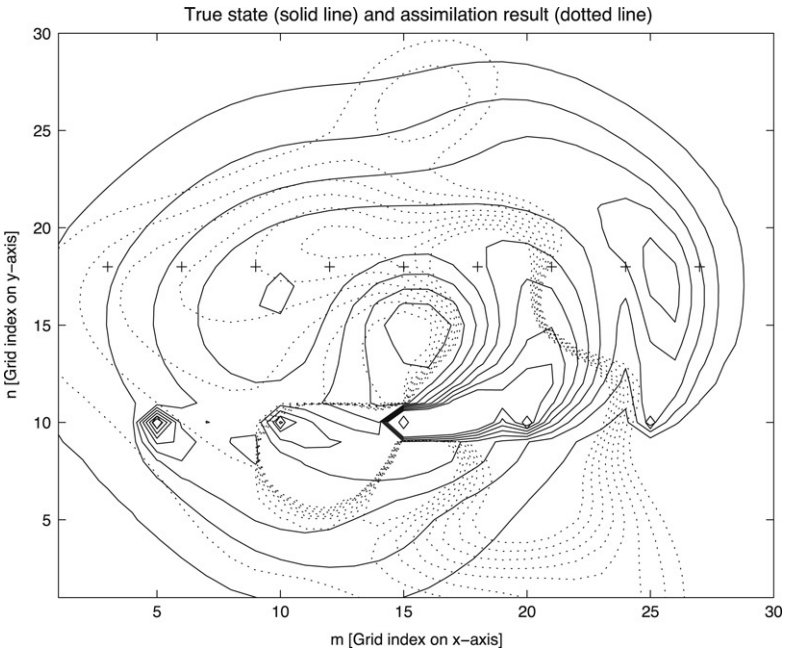

Fig. 7. The concentrations calculated using an ENK filter with $(q, N) = (5, 30)$ at time step $k = 100$ (Case I).

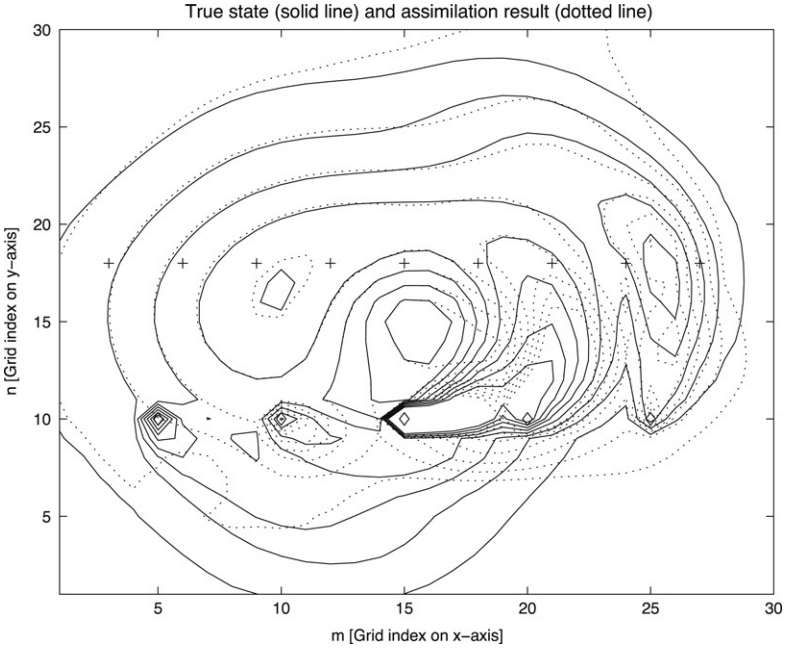

Fig. 8. The concentrations calculated using an ENK filter with $(q, N) = (25, 30)$ at time step $k = 100$ (Case I).

Thus, we will distinguish on the whole domain two straight lines of points: a horizontal line containing only emission points placed in the lower part of the domain and another horizontal line constructed only with measurement points situated in the upper part of the domain. The situation where the distance between the two horizontal lines is small is analyzed in Case I. The increase of this distance (the distance between measurements and emission points) is studied in Case II. The motivation of the introduction in our analysis of Case I and Case II comes from real-life applications, when not the whole area of the space domain is qualified as accessible for measurement points.

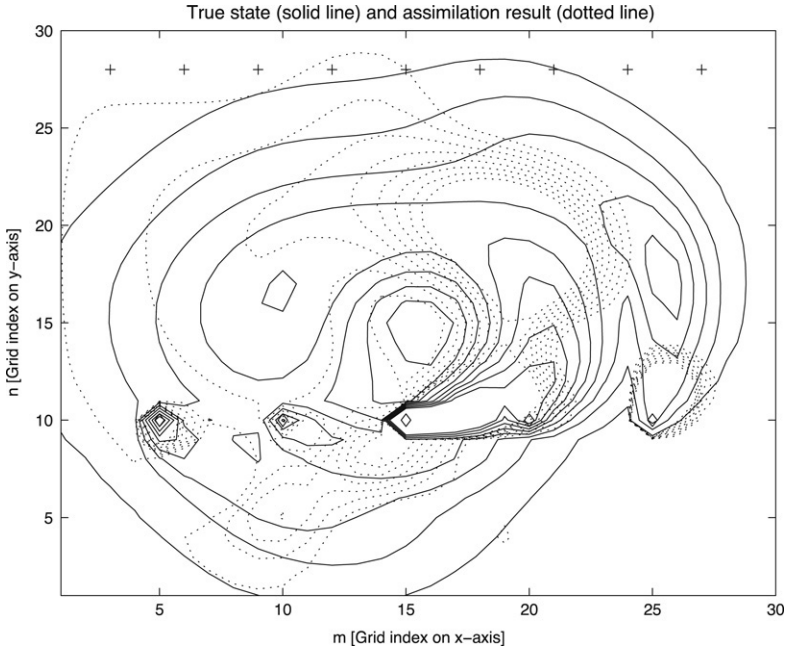

Fig. 9. The concentrations calculated using an ENK filter with $(q, N) = (5, 30)$ at time step $k = 100$ (Case II).

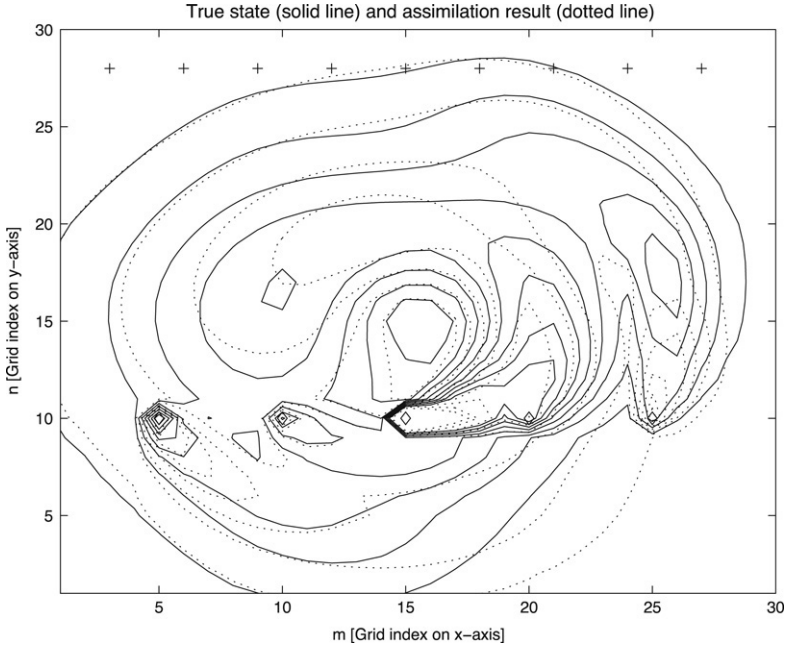

Fig. 10. The concentrations calculated using an ENK filter with $(q, N) = (25, 30)$ at time step $k = 100$ (Case II).

As a model under investigation, we consider the 2-D advection–diffusion equation:

$$\frac{\partial \mathbf{c}}{\partial t} = \nu \left(\frac{\partial^2 \mathbf{c}}{\partial x^2} + \frac{\partial^2 \mathbf{c}}{\partial y^2} \right) - u \frac{\partial \mathbf{c}}{\partial x} - v \frac{\partial \mathbf{c}}{\partial y}, \tag{19}$$

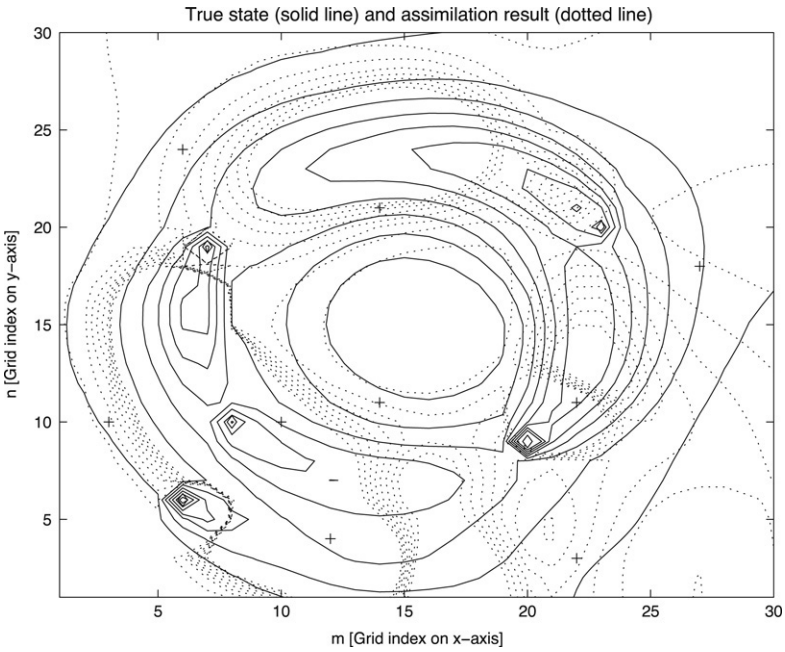

Fig. 11. The concentrations calculated using an ENK filter with $(q, N) = (5, 30)$ at time step $k = 100$ (Case III).

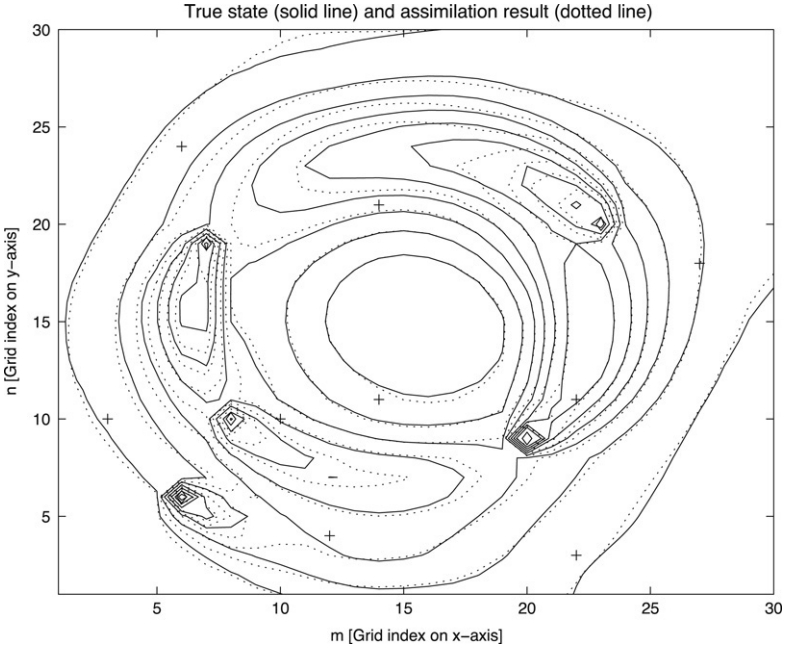

Fig. 12. The concentrations calculated using an ENK filter with $(q, N) = (25, 30)$ at time step $k = 100$ (Case III).

with a square domain and zero initial conditions. The concentration at the boundary is zero for inflow. We set the diffusion coefficient $v = 0.2$ and define the wind velocities u and v as follows:

$$u = -vel.scale * (y_{\text{grid}} - yc_{\text{vortex}}), \qquad v = vel.scale * (x_{\text{grid}} - xc_{\text{vortex}}).$$

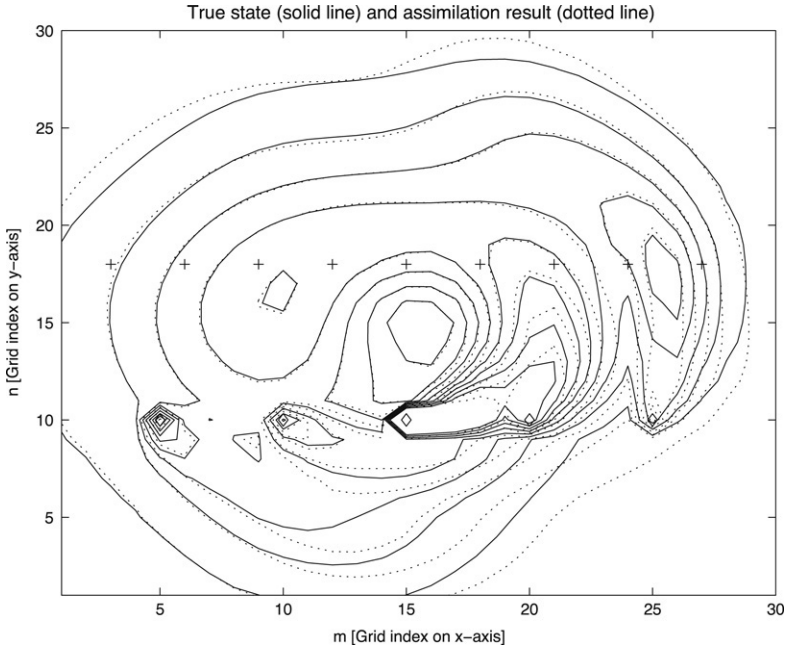

Fig. 13. The concentrations calculated using a POEN filter with $(q, N) = (5, 30)$ at time step $k = 100$ (Case I).

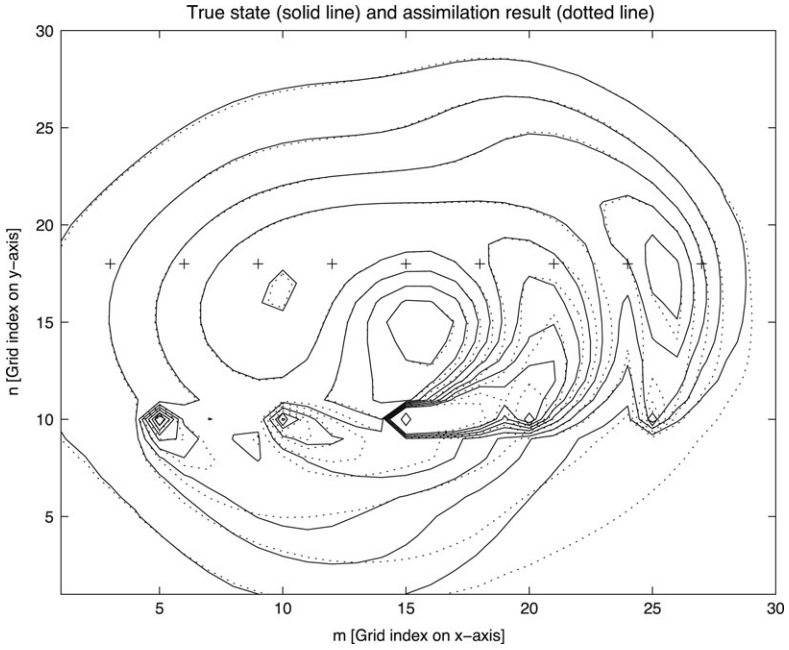

Fig. 14. The concentrations calculated using a POEN filter with $(q, N) = (25, 30)$ at time step $k = 100$ (Case I).

Here, *vel.scale* denotes the velocity scale computed at the value 0.06 $(2 * 0.9$ divided by 30, the maximum number of grid points in one direction), $(x_{\text{grid}}, y_{\text{grid}})$ denotes the current grid point, and by $(xc_{\text{vortex}}, yc_{\text{vortex}})$ we specify the center of the vortex.

We used a backward Lagrangian scheme to discretize these equations on a 30×30 grid. The velocity field is considered known and constant in time with a vortex located in the middle of the domain.

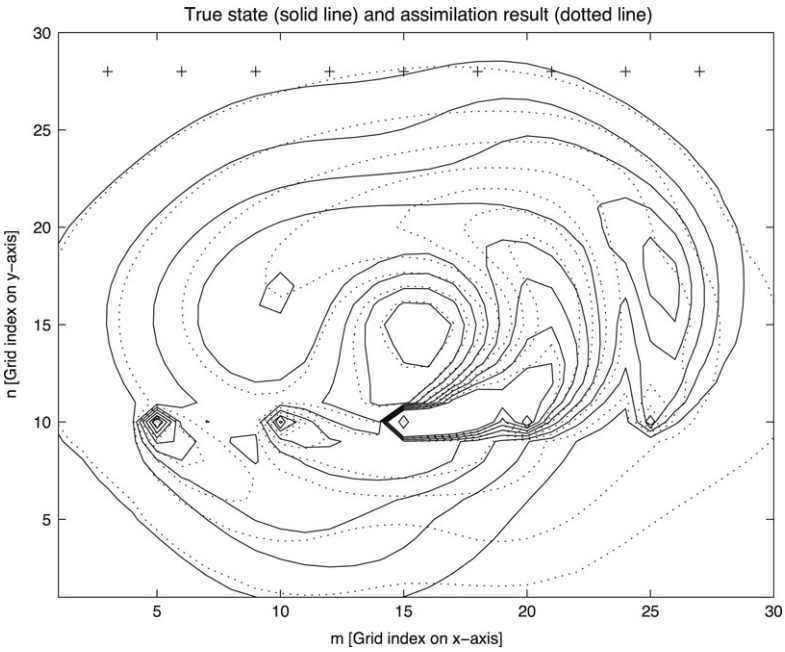

Fig. 15. The concentrations calculated using a POEN filter with $(q, N) = (5, 30)$ at time step $k = 100$ (Case II).

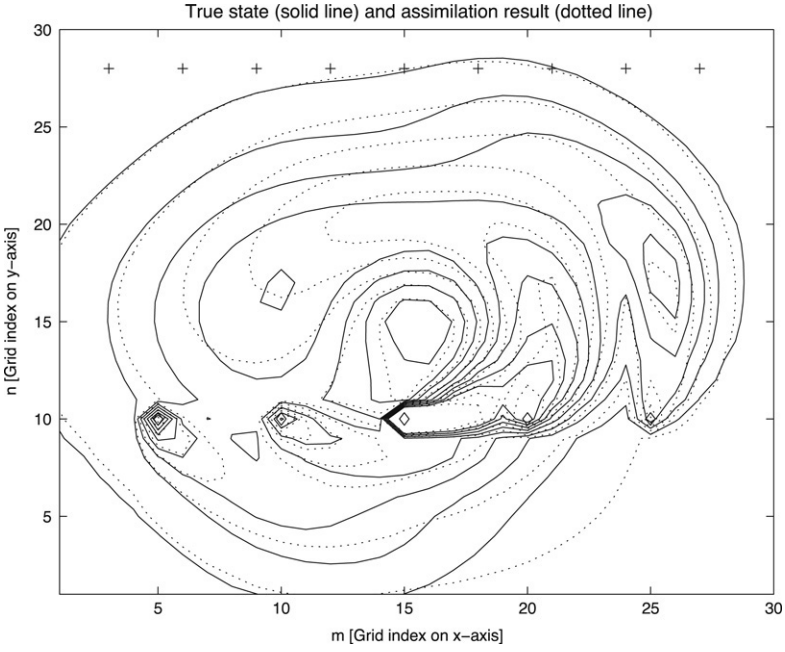

Fig. 16. The concentrations calculated using a POEN filter with $(q, N) = (25, 30)$ at time step $k = 100$ (Case II).

In Cases I and II a reference solution was generated by inserting constant emissions at five pollution locations, equally spaced on a straight horizontal line at the grid cells $\{(5, 10), (10, 10), (15, 10), (20, 10), (25, 10)\}$. The increase of concentration per timestep for these location was set to $\{0.2, 0.1, 0.1, 0.2, 0.2\}$, respectively.

In the same manner, the reference solution for Case III was generated by inserting constant emissions at grid cells $\{(6, 6), (8, 10), (20, 9), (7, 19), (23, 20)\}$. The increase of concentration per timestep for these location was maintained at the same values as in Cases I and II.

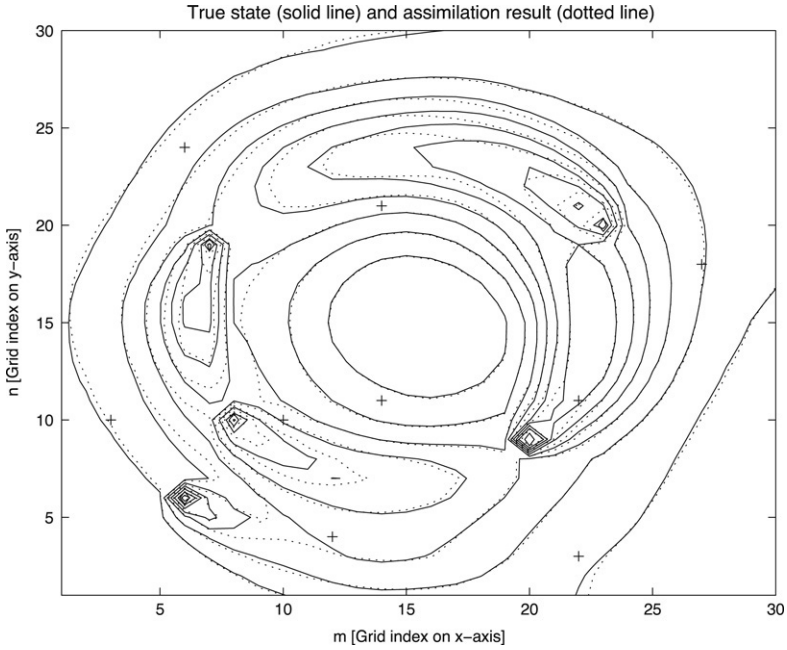

Fig. 17. The concentrations calculated using a POEN filter with $(q, N) = (5, 30)$ at time step $k = 100$ (Case III).

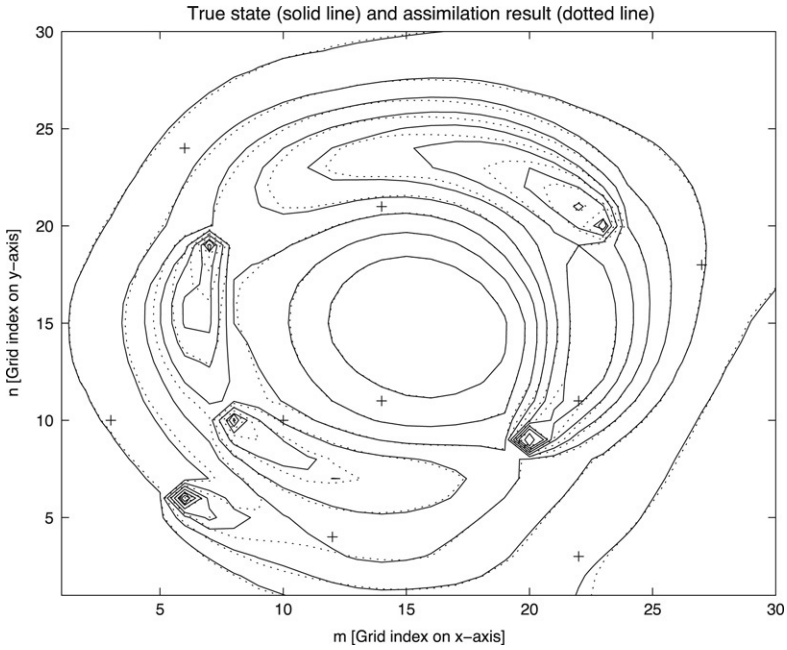

Fig. 18. The concentrations calculated using a POEN filter with $(q, N) = (25, 30)$ at time step $k = 100$ (Case III).

We want to point out that the pollution locations within a certain analysed case (I, II or III) have different mean pollution rates, but identical values of these rates for the corresponding pollution points in all three cases. Thus, the third point (with the coordinates (15, 10) in Cases I and II, and (20, 9) in Case III) has the biggest mean pollution rate among all the five points. It follows, in this order, the second, the fifth, the first and finally, the fourth points. We can remark in Figs. 1–24 that the pollution location with the biggest mean pollution rate is surrounded by many streamlines shifted to the right by the wind velocity.

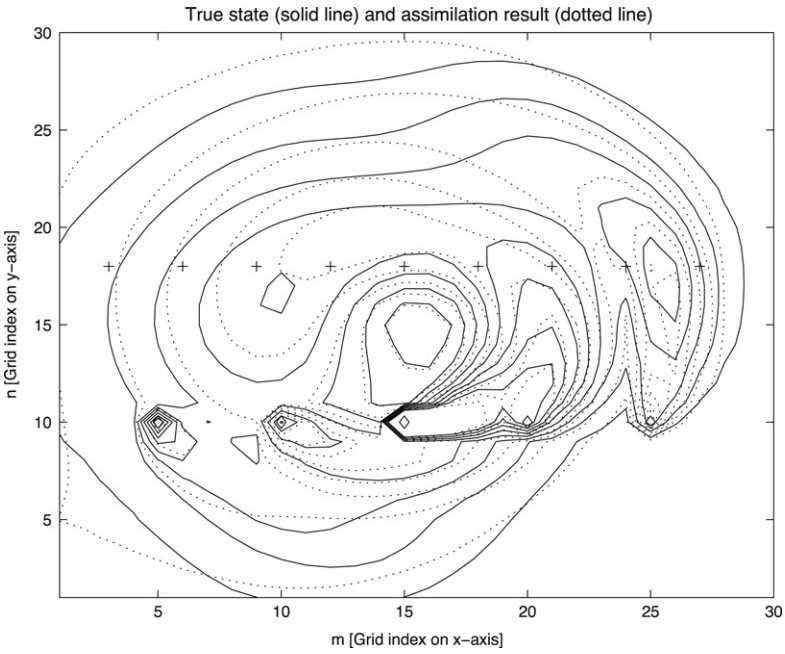

Fig. 19. The concentrations calculated using a COFFEE filter with $(q, N) = (5, 30)$ at time step $k = 100$ (Case I).

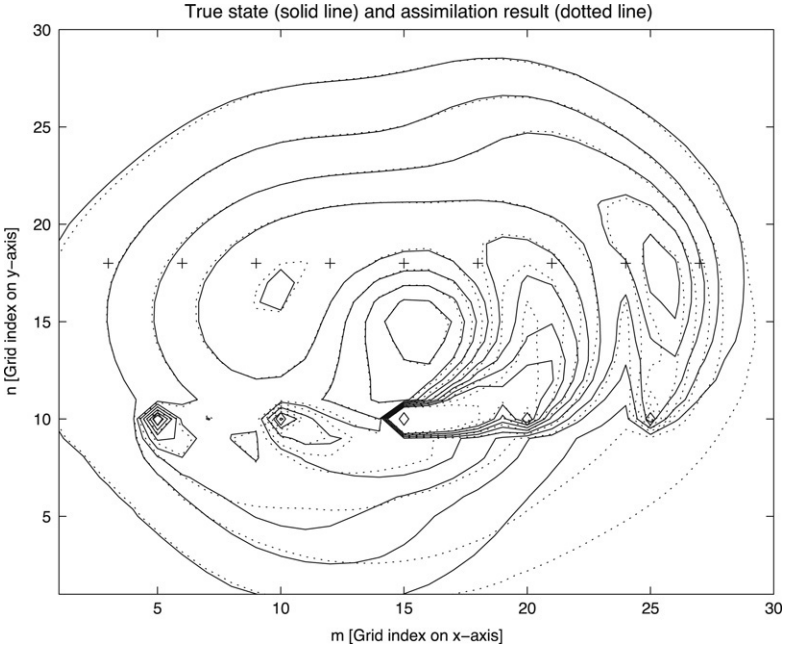

Fig. 20. The concentrations calculated using a COFFEE filter with $(q, N) = (25, 30)$ at time step $k = 100$ (Case I).

In Case I, the measurements were generated from simulated true concentrations, which were computed by adding fluctuations to the mean emissions according to $\tilde{z}_j(k+1) = \gamma_j \tilde{z}_j(k) + z_j(k)$, with independent Gaussian white noise processes with $E\{z_j(k)\} = 0$ and $Var\{z_j(k)\} = 1$. The index j refers to measurement locations (3, 18), (6, 18), (9, 18), (12, 18), (15, 18), (18, 18), (21, 18), (24, 18), (27, 18), and γ_j is a decay per step corresponding to the measurement location. The 9 measurement locations are placed on a horizontal line at a distance (measured vertically) of 8 grid points from the line defined by the pollution locations.

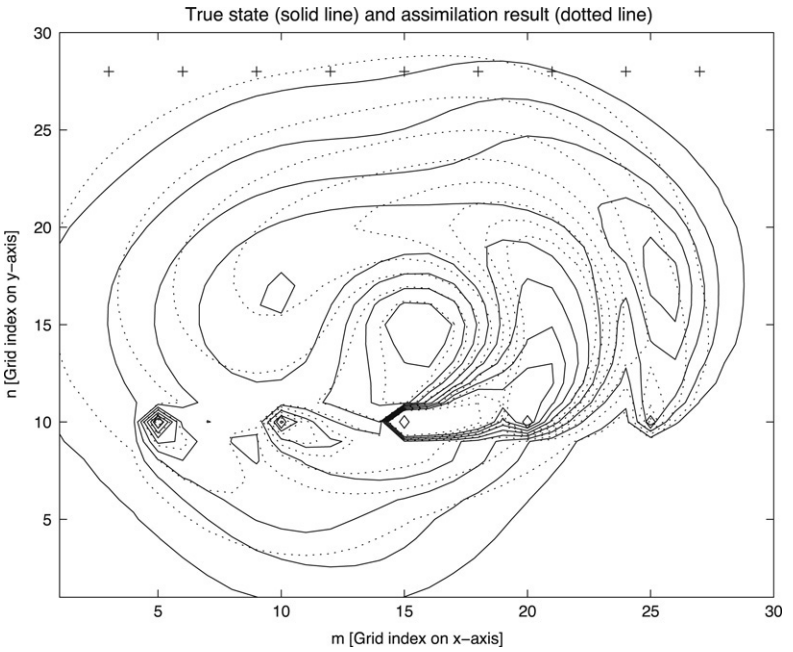

Fig. 21. The concentrations calculated using a COFFEE filter with $(q, N) = (5, 30)$ at time step $k = 100$ (Case II).

The measurement locations in Case II are defined by the following coordinates of the grid cells: (3, 28), (6, 28), (9, 28), (12, 28), (15, 28), (18, 28), (21, 28), (24, 28), (27, 28). Therefore, the distances between the emission points and measurement locations are now greater than that in Case I (the distances of 18 grid points measured vertically).

In the last case under study (Case III), the measurement locations are given by the grid points: (3, 10), (12, 4), (27, 18), (14, 11), (22, 3), (10, 10), (14, 21), (22, 11), (6, 24). Finally, white observational noise with a variance of 0.1 was added to the true concentrations. To compare the performance of the different filters with each other, the root mean square (RMS) errors were computed:

$$RMS = \sqrt{\frac{1}{N_x^2 N_t} \sum_{m,n,k} (\mathbf{c}_{m,n}(k) - \hat{\mathbf{c}}_{m,n}(k))^2}, \tag{20}$$

where $\mathbf{c}_{m,n}(k)$ are the exact generated concentrations and $\hat{\mathbf{c}}_{m,n}(k)$ are the estimates computed, N_x is the number of gridpoints in one direction and N_t is the number of timesteps.

If the RMS errors of all experiments are compared (see Tables 1–4), the RRSQRT algorithm seems to have a robust behaviour for this particular application. The filter provides an accurate and constant result at a level of required model evaluations where the other algorithms still suffer from random fluctuations. Even for small numbers of modes, the results are more accurate than what could be achieved with an ENKF approach with a comparable ensemble size. These results show that the convergence of the RRSQRT filter is much faster than the convergence of the ensemble filter. In Figs. 1–24 the concentration fields of the truth-run and the reference-run are shown after $N_t = 100$ timesteps. The $+$ signs indicate the measurement locations and the diamond-signs the locations of the emissions. It can be noticed clearly that the true fields are perturbed with time-varying fluctuations, while the reference solutions only contain a steady emission which is advected and spreading smoothly.

In Figs. 1–6 we show the concentrations calculated using an RRSQRT filter at the final time $k = 100$ for 30 ensemble members. We present the assimilation results both for 5 and 25 modes as well. We remark that the RRSQRT filter for the settings $(q, N) = (5, 30)$ do not perform well, creating spurious streamlines inside the domain. Its RMS error is 1.447 for Case I, 1.068 for Case II and 0.9295 for Case III (see Table 1). Increasing the number of modes to $q = 25$, we obtain better results, with RMS errors diminished at the values 0.482, 0.562 and 0.351 for the Cases I, II and III, respectively. An improved assimilation result can be seen in Figs. 2, 4 and 6. We notice that the RMS error obtained in Case I is smaller than the RMS error in Case II. This fact strongly motivates the decision that the positions

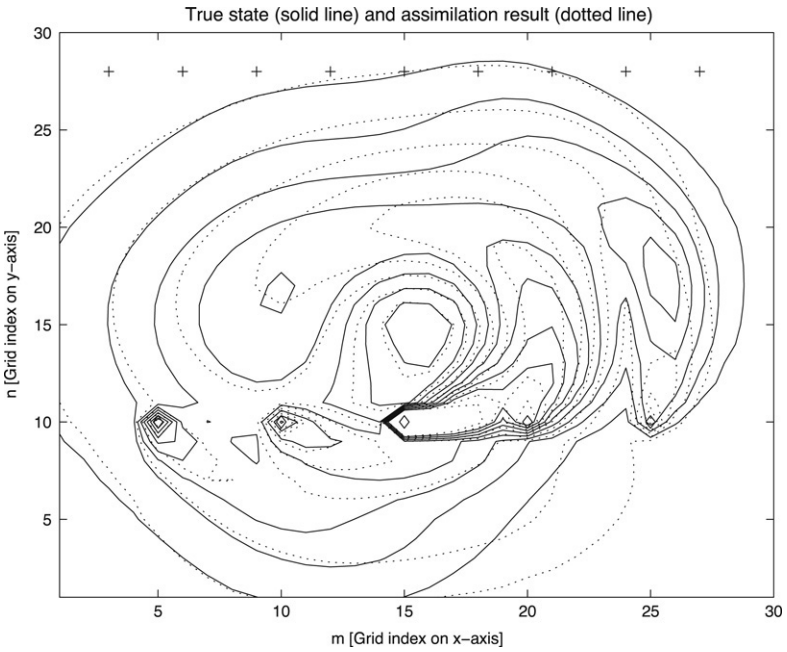

Fig. 22. The concentrations calculated using a COFFEE filter with $(q, N) = (25, 30)$ at time step $k = 100$ (Case II).

Table 1
Numerical results using RRSQRT filter — RMS errors and standard deviation (STD) values for concentration and system noise

Modes q	Ensemble N	RMS conc. (Case I)	RMS conc. (Case II)	RMS conc. (Case III)	STD conc. (Case I)	STD conc. (Case II)	STD conc. (Case III)
5	30	1.4479	1.0680	0.9295	0.1752	0.2080	0.1578
10	30	1.4941	0.6154	0.6582	0.2715	0.2875	0.2533
15	30	1.4412	0.6129	0.3970	0.3320	0.3587	0.3082
20	30	1.0725	0.6652	0.3500	0.3842	0.4171	0.3197
25	30	0.4826	0.5620	0.3519	0.4102	0.4599	0.3225
30	30	0.4356	0.5822	0.3522	0.4144	0.4785	0.3231
Modes q	Ensemble N	RMS noise (Case I)	RMS noise (Case II)	RMS noise (Case III)	STD noise (Case I)	STD noise (Case II)	STD noise (Case III)
5	30	6.8737	3.0845	4.9441	1.4422	1.2871	1.4431
10	30	4.7941	2.1529	2.9814	1.6663	1.4653	1.7231
15	30	3.1432	2.1278	2.1266	1.8626	1.6315	1.9407
20	30	3.3308	2.2157	2.0632	1.9631	1.8059	1.9663
25	30	2.0726	2.0803	2.0653	1.9981	1.9503	1.9706
30	30	2.0533	2.1084	2.0655	2.0022	2.0151	1.9715

of the measurement locations are to be settled as closely as possible (or in any case, not far) to the pollution locations. The same decrease of RMS errors is found when one uses POEN and COFFEE filters (see Tables 3 and 4).

The assimilation results of the concentrations calculated with The ENK filter are contained in Figs. 7–12. The ensemble filter suffers from statistical noise due to the use of a random number generator. Large values of RMS errors are obtained for small number of ensemble members (see Table 2).

In Figs. 13–24 the results for the Partially Orthogonal Ensemble Kalman filter and Complementary Orthogonal Filter For Efficient Ensembles are presented at the final time $k = 100$. We observe that both the POENK filter and the COFFEE algorithm have a more robust behaviour for small ensemble sizes than RRSQRT and ENSEMBLE filters.

Table 2
Numerical results using ENSEMBLE filter — RMS errors and standard deviation (STD) values for concentration and system noise

Ensemble N	RMS conc. (Case I)	RMS conc. (Case II)	RMS conc. (Case III)	STD conc. (Case I)	STD conc. (Case II)	STD conc. (Case III)
5	6.0610	1.4746	4.2539	0.1742	0.2693	0.1146
10	1.4567	0.7936	0.7806	0.2002	0.3147	0.1441
15	0.7683	0.6937	0.6210	0.2585	0.3577	0.1877
20	0.7284	0.7651	0.5692	0.2813	0.3614	0.1958
25	1.0245	0.6061	0.4529	0.2940	0.3772	0.2102
30	0.9070	0.7215	0.4410	0.3079	0.4112	0.2405
Ensemble N	RMS noise (Case I)	RMS noise (Case II)	RMS noise (Case III)	STD noise (Case I)	STD noise (Case II)	STD noise (Case III)
5	31.598	5.0479	18.245	1.6084	1.7061	1.3832
10	7.2784	2.6368	6.0983	1.6648	1.7845	1.5585
15	3.4405	0.3822	2.8188	1.7767	1.9152	1.6857
20	2.7913	2.3777	2.6913	1.7554	1.8361	1.6657
25	3.6213	2.1128	2.2937	1.7718	1.8672	1.6922
30	3.0147	2.3221	2.2190	1.9056	1.9779	1.7992

Table 3
Numerical results using POEN filter — RMS errors and standard deviation (STD) values for concentration and system noise

Modes q	Ensemble N	RMS conc. (Case I)	RMS conc. (Case II)	RMS conc. (Case III)	STD conc. (Case I)	STD conc. (Case II)	STD conc. (Case III)
5	30	0.5110	0.6166	0.3998	0.1869	0.2112	0.1663
10	30	0.4603	0.6192	0.3912	0.2772	0.2907	0.2584
15	30	0.4884	0.6064	0.3862	0.3457	0.3620	0.3088
20	30	0.4527	0.5909	0.3722	0.3960	0.4195	0.3199
25	30	0.4563	0.5940	0.3749	0.4120	0.4621	0.3223
30	30	0.4514	0.5919	0.3755	0.4157	0.4799	0.3229
Modes q	Ensemble N	RMS noise (Case I)	RMS noise (Case II)	RMS noise (Case III)	STD noise (Case I)	STD noise (Case II)	STD noise (Case III)
5	30	2.2615	2.1902	2.2381	1.2916	1.2646	1.3661
10	30	2.1610	2.1882	2.2097	1.5472	1.4460	1.6988
15	30	2.1684	2.1820	2.1870	1.7782	1.6137	1.9354
20	30	2.1681	2.2001	2.1710	1.9368	1.7919	1.9675
25	30	2.1706	2.1720	2.1735	1.9984	1.9428	1.9707
30	30	2.1641	2.2021	2.1742	2.0024	2.0137	1.9715

4. Concluding remarks

In this study, four different low-rank filters have been implemented in a 2-D advection–diffusion model; these are based on factorization (RRSQRT filter), ensemble statistics (ENKF), or on hybrid approaches (the POENKF combining an RRSQRT and ENKF filter, and its variant the COFFEE filter). All four methods were found to be suitable for assimilating data with stochastic varying emissions. The ensemble filter suffers from statistical noise due to the use of a random number generator; the results still show a large spread where an RRSQRT filter with comparable costs has already converged. As a consequence, the POENKF filter also suffers from the same statistical noise in its ENKF part. Due to the fast convergence and accurate results reached with the RRSQRT filter, the benefit of additional random directions in the gain of the POENKF is limited. For comparable costs, the RRSQRT filter produces stable and more accurate results than ENKF or POENK and COFFEE filters.

Table 4
Numerical results using COFFEE filter — RMS errors and standard deviation (STD) values for concentration and system noise

Modes q	Ensemble N	RMS conc. (Case I)	RMS conc. (Case II)	RMS conc. (Case III)	STD conc. (Case I)	STD conc. (Case II)	STD conc. (Case III)
5	30	0.5511	0.6140	0.4384	0.1888	0.2108	0.1656
10	30	0.5042	0.5094	0.3989	0.2790	0.2885	0.2605
15	30	0.4954	0.5084	0.3604	0.3467	0.3594	0.3062
20	30	0.4361	0.6003	0.3550	0.3978	0.4185	0.3211
25	30	0.4495	0.5361	0.3539	0.4142	0.4626	0.3232
30	30	0.4351	0.5550	0.3545	0.4167	0.4803	0.3236
Modes q	Ensemble N	RMS noise (Case I)	RMS noise (Case II)	RMS noise (Case III)	STD noise (Case I)	STD noise (Case II)	STD noise (Case III)
5	30	2.1453	2.1760	2.2749	1.3203	1.2759	1.3847
10	30	2.0803	2.0248	2.1539	1.5623	1.4515	1.7192
15	30	2.0971	1.9750	2.0737	1.7771	1.6190	1.9268
20	30	2.0797	2.5000	2.0757	1.9317	1.7927	1.9692
25	30	2.0683	2.0615	2.0718	1.9997	1.9429	1.9720
30	30	2.0642	2.0977	2.0691	2.0030	2.0144	1.9724

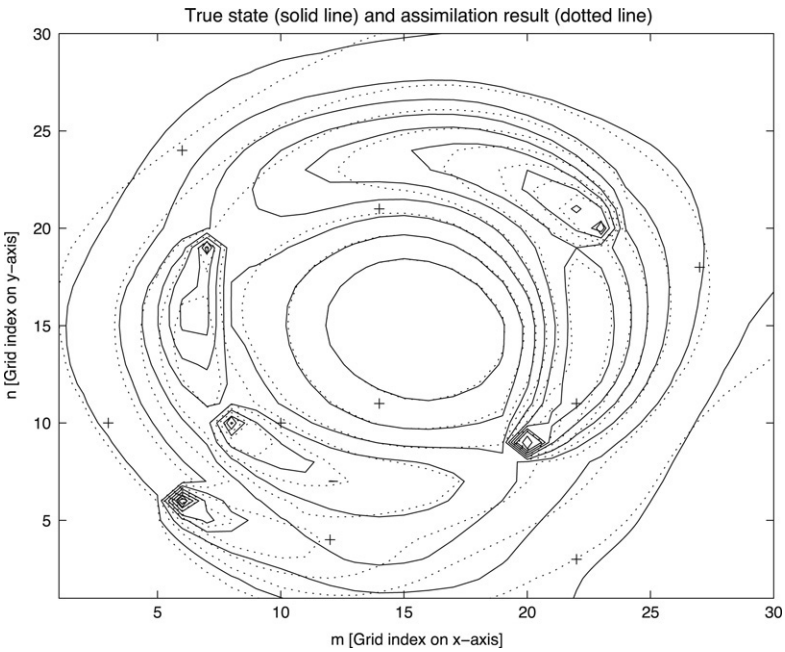

Fig. 23. The concentrations calculated using a COFFEE filter with $(q, N) = (5, 30)$ at time step $k = 100$ (Case III).

The results also indicated that it is favourable for data assimilation that the observation points be located as closely as possible to the emission points, since the RMS errors of all experiments showed a decrease of their values when this was done. In the near future, we intend to study the behaviour of such filters when one takes into consideration a vortex (or several vortices) with time depending positions inside the assimilation domain, thus capturing much more accuracy the actual features of complex atmospheric turbulences.

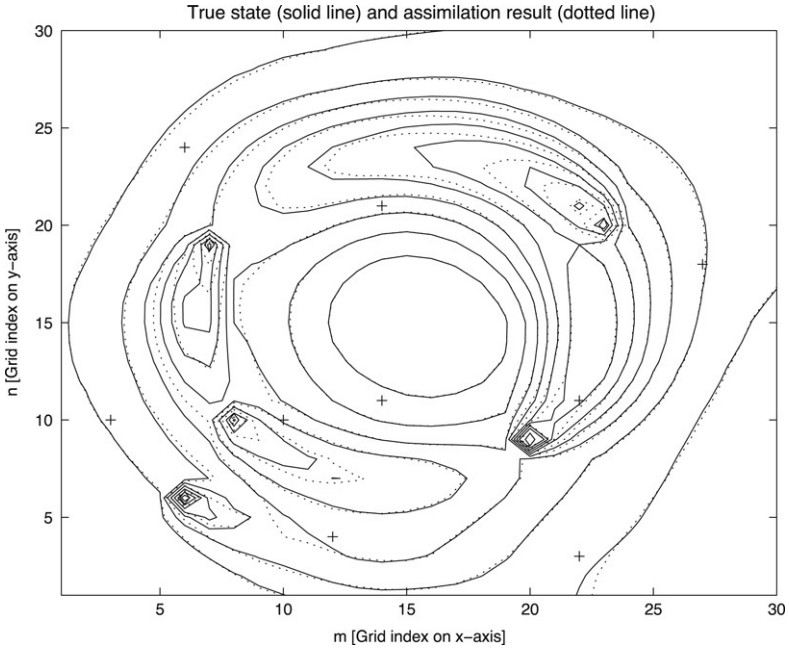

Fig. 24. The concentrations calculated using a COFFEE filter with $(q, N) = (25, 30)$ at time step $k = 100$ (Case III).

References

[1] R.E. Kalman, A new approach to linear filter and prediction theory, J. Basic Eng. 82D (1960) 35–45.
[2] M. Ghil, P. Malanotte-Rizzoli, Data assimilation in meteorology and oceanography, in: Advances in Geophysics, vol. 33, Academic Press, San Diego, California, 1991, pp. 141–266.
[3] G.J. Bierman, Factorization Methods for Discrete Sequential Estimation, in: Mathematical in Science and Engineering, vol. 128, Academic Press, New York, 1977.
[4] G. Evensen, Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics, J. Geophys. Res. 99 (C5) (1994) 10143–10162.
[5] G. Evensen, P.J. van Leeuwen, Assimilation of Geosat altimeter data for the Agulhas current using the ensemble Kalman filter with a quasi-geostrophic model, Mon. Weather Rev. 124 (1996) 85–96.
[6] P.L. Houtekamer, H.L. Mitchell, Data assimilation using an ensemble Kalman filter technique, Mon. Weather Rev. 126 (1998) 796–811.
[7] A.W. Heemink, A.J. Segers, Modeling and prediction of environmental data in space and time using Kalman filtering, Stoch. Environ. Res. Risc Asses. 16 (3) (2002) 225–240.
[8] A.W. Heemink, M. Verlaan, A.J. Segers, Variance reduced ensemble Kalman filtering, Mon. Weather Rev. 129 (7) (2001) 1718–1728.